ISOMETRIC: A framework for identifying perceived similarity using abstract structures and EEG

1st Septimiu-Călin Bodică
Babeș-Bolyai University
Cluj-Napoca, Romania
septimiu.bodica@stud.ubbcluj.ro

***Abstract*—Using EEG to conduct scientific investigation is prone to issues because of unmeasurable parameters such as individual traits, preconditions and context. It is often a problem that a study is not translatable to a different setting than the one in which the measurements were taken. This study introduces a framework: ISOMETRIC, a hand-drawing human-in-the-loop system that makes use of algebraic groups: reflection and rotation groups of representation theory, for understanding the perceived relationship between objects in a scene. The mathematical notions are applied to hand-drawing tasks on a digital tablet, where we show that EEG data annotated with algebraic group information describing the drawing task can serve as a tool to classify and interpret EEG signals. A pilot study is conducted to show the framework's ability to cluster signals across multiple experiments.**

***Index Terms*—human-computer interaction, biofeedback, EEG, representation theory, group theory**

I. Introduction

Biofeedback applications combine sensing domains which measure different parameters of subjects as they perform various tasks in order to study how the task influences the subjects. A biofeedback application incorporates useful data and transmits some of the data back to the user through various methods. The data can also be collected for the purpose of eliminating undesired or irrelevant data that is part of the system. A biofeedback study motivated by correctly formed hypotheses that would have a high likelihood of being correct can be seen as the basis of any biofeedback system. ISOMETRIC framework is a hand-drawing application that combines EEG data with the process of drawing on a digital tablet, and uses the principles of representation theory to reduce the complexity of the data that is analyzed.

In this study we aim to show how the language of algebraic groups is related to the problem of designing a hand drawing biofeedback application. The task of the user is to draw along the recommendations of the ISOMETRIC engine to piecewise construct a regular tiling or quasiperiodic tiling across the whole board. A tiling is a set of regular polygons that are connected to each other at the edges and constructed symmetrically by repeated operations. This activity would allow the subject to closely interact with abstract representations, and as habituation occurs, to identify how each individual understands the abstract representations that describe the geometric structures locally as well as globally. The mathematical properties of the tiling allows the framework to tackle one of the largest problems in human-computer interaction, that of having the ability to generalize results and take the findings out of the controlled setting. Once the user's biometric signals are profiled with respect to the perception of abstract representations, the framework can know what signal patterns to look for when the subject perceives an environment which evokes the perception of the same abstract representation. This means that ISOMETRIC isolates specific patterns that the human brain prefers, and allows using the insights to aid in tasks such as optimizing environments and enabling user interfaces to directly interact with the user's pattern recognition abilities.

Human-computer interaction (HCI) is concerned with the similarities and dissimilarities between the human and the computer. The two are viewed as systems that communicate and mutually exchange information. Considering biofeedback applications, the experience of a computer user interacting with a computer through the use of a screen and classical human input devices gives a more controlled setting than performing the measurements "in the wild". Measurements about the parameters of the computing environment and the signals emitted by the human body are refined and they are used in proving or disproving hypotheses about how the two systems: the human and the computer, influence each other.

We suggest it might be possible for BCI applications to include information about the pattern recognition capabilities of the human brain in their design. Objects in a virtual environment are represented as a scene graph where each object is composed of smaller objects. An observer will observe the scene from a single point of view, therefore the objects will be clustered together in some regions and sparse in other regions. In order to prevent brain fatigue, optimize scene layout, and best interact with the virtual environment, the layout of the objects in the scene could be distributed according to a rules based system that uses reflection and rotation groups. This study aims to answer the question of how a virtual environment would best detect and interact with the user's pattern recognition system.

The problem that can arise is that the space of viable hypotheses regarding what causes changes in the neural signals is very large. For example there could be problems with associating signals situated close to the temporal lobe to those on the central area because there might be hidden variables such as obstructed or unmeasured regions on which the signals depend. For this reason a decision support system or proof assistant may be better suited to probe the realm of possibilities. [1], [2]

Recently BCI have been used with more success for communicating with disabled patients in the form of neurofeedback systems that co-adapt with a learning patient in order to use raw brain activity combined with tracking a sound signal to produce natural language text. [3] Because of the structured nature of brain connectivity, electric activity measures can be interpreted differently depending on the location where they occur. [4] The way in which signals propagate through the brain is hard to predict and nonlinear. Some theories about how signals propagate through the brain are compatible with the concept of receptive fields which have been investigated in separate research, and offers a degree of explainability for subjects interacting with computer screens. [5] In the case of visual perception, the raw optical input arrives in the hindbrain, or occipital area, and gets diffused along the temporal and central axes, eventually having the potential of reaching the frontal lobe.

For the receptive fields of the visual cortex, one could expect the electrodes T3, T4, T5, T6 to activate for handwriting because they are located close to the temporal lobe, and more predominant C3, CZ, C4 activity for drawing faces. The most raw input is found in the occipital region. The apparent justification for such a hypothesis is that the cortical homunculus associated to the facial nerves has been identified to be closer to the central region of the electrode map. [6] At the same time, both of these regions are in proximity to regions associated with high order visual areas. Thus the signals are dependent on the sensory stimuli. The signal's localization can also depend on the patient's profile, and personal characteristics. The problem remains that the space of hypotheses to test with respect to the neural signals is very large, and are often explored using bayesian inference on large probabilistic computational models. [1], [2]

II. Related work

Most of the work presented in the framework reduces the interaction with a computer system through a digital tablet to a set of elementary operations that can be found in any type of computer application. The objective can be viewed as the study of a generalized context in order to obtain empirical evidence of how biometric signals relate to human-computer interaction. One of the tools used is the mathematical language found in linear algebra, group theory, geometry. Because the environment which the application runs in is a Turing machine built on *von Neumann* architecture, there is an implicit mathematical toolset for discussing the complexity and computability of the application. Techniques from formal methods, automata theory, static analysis may be seen as related to the nature of the tilings and the computational environment. The computational environment in large part constitutes the setting of the study. The language used for generating the tilings have a similar nature to some formal grammars. Tilings are well categorized and because the rules of their constructions are strict the theory does not allow invalid constructions. Formal methods encounter problems when they are applied to the highly nonlinear and complex human brain, or even some larger neural network models. The mathematical models described in relation to tilings can only outline a problem that is intrinsically very complex. By analogy, one can consider the problem of the undecidability of program termination, or quasiperiodic tilings.

To illustrate a potential bridge between the simple mathematical description of the task and and the domains of HCI, consider the classical problem of the brain-computer interface, which has been a subject of research since the 1930s. A BCI has to solve a multitude of tasks: abnormality detection, signal classification, pattern detection, predictions on a time window, generating synthetic signals and interpreting the data with respect to other sensor modalities, such as heart rate or muscle movement. To solve all of these tasks, the analog signals must be converted to digital signals and then stored. Using a computer with an operating system has the advantage that the solution provided can follow design patterns and use architectures that are easy for the creator to manipulate. A common architecture that simplifies development of complex reactive systems is the event-driven architectures.

Human-readable data structures can be used in specifying the program's behavior. The events and data structures have a nested or hierarchical representation, because some of the tasks are dependent on the complete execution of other tasks situated lower in the hierarchy. Such hierarchical designs tend to form a so-called *pipeline*, where the raw signal is transformed successively and branched into parallel processing steps. The human agent can be considered to be a part in the pipeline, or part of a more complex human-in-the-loop system. The only way to cope with the complexity of nonlinear systems is to limit their input, which is the *interface* with the environment, or constrain the behavior of the system instead, like reducing the number of output channels. Adding constraints enables some properties of the system to be derived from theory and allows the systems to even be simulated to a certain degree.

To detect abnormalities, the signals are analyzed in sequences of batches of a given length of time. Statistics are extracted at the granular level of individual time windows, or at a more general level the correlations between properties of the time windows can be considered. The field of digital signal processing is large, and includes fields such as spectral analysis, Fourier analysis, and the study of nonlinear systems. By comparison to digital signals, analog signals are continuous in the time domain. An analog signal can be very precise until the threshold where the noise is inseparable from the communicated signal.

Sometimes specialized hardware is used to convert the analog signal which is obtained from the mechanical or electrical sensor into a digital signal, without having to assign this responsibility to the computer, and this is done using an analog-to-digital converter (ADC). The inverse task is performed with a digital-to-analog converter, but in the processing pipeline we are interested in the digital signal. If the instrument possesses an ADC, then the signal that is obtained from the instrument is already split into discrete values at a certain sampling rate ω measured in Hz. The discrete set of values can be reconstructed from the samples using numerical approximations. Two of the most important representations of signals which are also compatible with computers are the STFT and the wavelet transform. The wavelet transform can be used to define a Hilbert basis of orthogonal wavelets which can be used to reconstruct the original signal. [7]

Predicting signals is not trivial. Supposing that the raw signal is used as input in a prediction pipeline, the prediction of the next time interval will possibly depend on the previous one and on statistical indicators that are correlated with the perceived environment. As habituation occurs the signals could change as a function of the biological factors or personal traits that are unique to each individual. Characteristics of different subjects might also appear as statistical indicators. Even a pipeline of hand-crafted features could contain errors outside a narrow set of constraints in which the system can be used. Whether using a machine learning model or a hand crafted signal processing pipeline based on theoretical facts, the problem is understanding the justification of the prediction, which is likely to be an abstract list of neurons weights in the case of neural networks with no interpretable meaning outside of the neural network itself, or a pipeline that fails to account for all of the variables of the environment. [8]

III. Model and methods

This chapter is structured in as follows: in the first part, the environment and tools are described. A mathematical description of the problem is given, then referenced to make arguments about software-related aspects such as data structures and design choices. Finally tilings introduced are shown to be a good choice of a drawing task in accord with the mathematically described problem domain. The framework presented should be thought of as a human-in-the-loop system which elicits a response of the pattern recognition system of the brain by using the most fundamental building blocks that can be offered by a classical virtual system which uses classical human-machine interaction devices.

A. Problem model

The central element of the application is the drawing board. This can be regarded to be the main component of the framework because it is the main source of kinematic data. Kinematic data can be considered the main stimulus which could evoke brain responses. The user's unique way of representing objects through hand drawing has been already studied, and different user traits have been shown to exist.

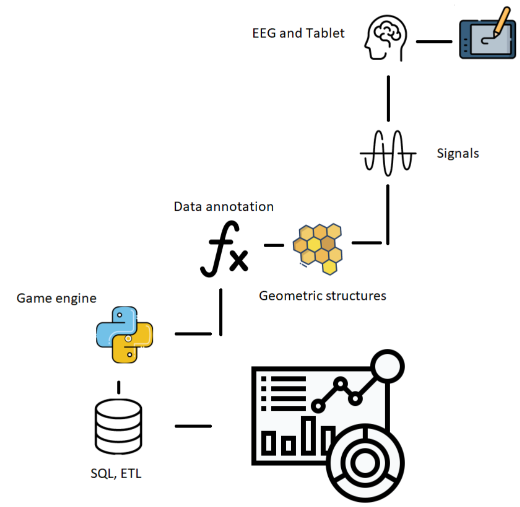

Fig. 1. Overview of the study setup. icons: Flaticon.com

[9] An argument for choosing hand drawing is that it is a subset of the process involved in handwriting, which is a task that employs cognition. What is omitted in drawing or writing is the phonetic aspects of speaking a language, which goes to show that humans internally represent concepts such as languages by integrating separate perspectives, like speaking, writing words, writing sentences, reading sentences into one whole. This is even more interesting when using languages composed of syllabic scripts mixed with multiple writing systems with numerous homographs, like Japanese and Kanji, in which the fundamental strokes compose larger units that have a meaning related to the meaning of the building blocks. The conclusions found in Morita et al. and Shibata et al. show the types of conclusions that one can make from natural language data, which is related to the context awareness of hand drawing data. [10], [11]

A drawing is composed of strokes, which is the interval of time between the moment when the pen is activated, and deactivated. Basic kinematic data comes with this such as velocity and acceleration. The shape drawn in a single stroke may be simple, for example a line, or complex, such as a set of curves or splines. Other dimensions of the stroke are the colour used, the thickness and shape of the brush, the orientations of the cursors, the symmetry axes of the cursors and their relative positions.

Let there also be another category of information about the strokes that is called second-order characteristics. These are as follows: order of the strokes, family of slopes, time of initiation of a stroke, and the relationship of a stroke to past and future strokes. Relationship between strokes is expanded on within our framework by using a recommender system for suggesting a future stroke. The recommender comes with different heuristics for proposing a new stroke, with special interest on forming self-similar patterns, or following a probability distribution based on the length of a sequence of recommendation, for example.

B. Mathematical description

First the data structures of importance throughout the discussion are specified.

The objective of the task is the tiling, composed of edges:

$$L = \{(p_i, p_j) \mid i, j \in \mathbb{N}, p \in P_c\}. \tag{1}$$

Denote the related set P_c of points within the tiling edges:

$$P_c = \{(s_x, s_y) \mid s \in \mathbb{R}\}, \tag{2}$$

such that

$$\forall l = (p_i, p_j) \in L, p_{ij} \in P_c. \tag{3}$$

A distinct set from P_c is P_s which is all points that can be sampled on the canvas.

$$P_s = \{(s_x, s_y) \mid s \in \mathbb{R}\}, \tag{4}$$

with $P_c \subset P_s$.

The current task is denoted as:

$$D = \{\forall s \in P_{c,s}\}. \tag{5}$$

It should be noted that the input of two distinct entities or game actors are involved in the final result: the user and the game engine. While the resulting state is a composite of P_c and P_s, both need to be stored separately by the game engine, for a few reasons. First of all, inputs could overlap, in fact it is the user's objective in certain game modes to perfectly draw over the game's input. To measure the similarity of the player's input and the optimal game engine's suggestion, the sources of the two sample points need to be distinguished.

This consideration is important when generalizing to multiple agents or decomposing the game engine's input into multiple recommendation styles.

Note that more or less, the points $s \in D$ correspond to pixels or paint strokes on the whiteboard.

In equation (4) we defined P_s which is a very general space of all points that can be sampled. We expand this into the set of strokes H_s that can be collected, where H stands for *history*, which ultimately belongs in the state machine. Let

$$H_s = \{\forall s = \{p_i, ..., p_j \mid p \in P_s\}\}, \tag{6}$$

with each stroke $s \in H_s$.

The following discussion is structured as follows: we describe the naturally occurring order structure induced by the temporal domain of the measurements. We then define a push-down automata (PDA) and a set of symbols belonging to its associated grammar $G = \{N, \Sigma, P, S\}$, which describes the inputs and has implications on *finite state machine* of the system. The symbol N describes the set of states, Σ the allowed symbols on the input tape, P the productions describing state transitions, and S the final states.

First of all, let:

$$t : X \longrightarrow \mathbb{R}, \tag{7}$$

be the time function, satisfying:

$$L, P, H \subset X. \tag{8}$$

We then have the natural order structure:

$$t(s_i) \leq t(s_j). \tag{9}$$

C. The state machine correspondence

The system is modelled by a push-down automata. Let M be a PDA:

$$M = \{Q, \Sigma, \Gamma, \delta, q_0, H_0, F\}, \tag{10}$$

with a set of states

$$Q = \{Init_i, Gen_i, Pred_i, Free_i \mid i \in \mathbb{N}\}. \tag{11}$$

The symbol Σ is the alphabet of symbols accepted on the input tape, Γ is the set of symbols accepted by the stack, δ the production rules that describe the transitions triggered by input symbols including stack behavior, q_0 the initial state, H_0 the initial stack symbols, and F the final states. These give the general classes of states but need not be complete in this description, because each individual state could be further decomposed and classified bringing little value to the discussion. Here, $Init_i$ is a class of initialization states when starting the program, Gen_i is a set of states used for generating the tiling objective, generating a new prediction or set of predictions, $Free_i$ is a cluster of states where the main mode of interaction is hand-drawing.

The PDA, M, is capable of switching game modes by parsing the input and triggering switches using the stack(s).To model all possible events within the PDA, as well as signals, the alphabet for the input is the following:

$$\Sigma = \{Point_{\delta xy}, K_{\updownarrow}, K_{\leftrightarrow}\}. \tag{12}$$

The associated set of rules δ of M are a set,

$$\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \longrightarrow Q \times \Gamma, \tag{13}$$

which makes use of the stack to implement switching logic, store the history of inputs H, and adjust the random variables of the probabilistic sampling χ_i such as when changing the number of predicted strokes and the frequency of the predicted strokes.

Consider $l \in L$ a possible prediction with $L \subset P_c \times P_c$. We would like to impose an order structure:

$$\begin{gathered} \tau_1 : l_1, l_2, ..., l_n \\ \Psi : L \times L \longrightarrow \{T/F\}, \\ \Psi(l_1, l_2) = m_{l_1} > m_{l_2}, \end{gathered} \tag{14}$$

where:

$$m_l = \frac{y_{p2} - y_{p1}}{x_{p2} - x_{p1}}, \quad p_{1,2} \in l,$$

and want to look for alternatives of Ψ where the comparison operator itself is a function:

$$g : L \times L \longrightarrow \{T/F\}, \Psi(l_1, l_2) = g(l_1, l_2).$$

For the inputs $K_\uparrow$ and $K_\rightarrow$ the state machine transitions into a batch prediction mode where multiple predictions are made. A prediction is a function of the order structure together with 2 random variables that describe the number k of strokes to predict, and the batches in relation to the time in which they will be provided.

$$P(\chi) = \begin{cases} 1, & \iff q = q_{full}, q \in Q \\ \frac{\lambda^k \cdot e^{-k}}{k!}, & \iff q = q_{timed}, q \in Q \end{cases} \tag{15}$$

where q_{timed} has a chance to give a prediction batch in a set of time intervals, but will always roll one of those intervals.

The order structure corresponding to their orientations τ_v and the map of their relative positions δ_v are all characteristics of the next prediction synthesized by the program. This information is calculated on the moment of providing this prediction as-needed, and not at the initialization phase. Therefore there is a map,

$$\phi : Q \times \Gamma \longrightarrow P \times \tau, \tag{16}$$

where the tuple-set,

$$\tau = \{(\tau_v, \delta_v, l) \mid l \in L\}, \tag{17}$$

is the set of possible current predictions mapped by ϕ.

Alternatively one should choose a mapping co-domain of ϕ s.t. additional information that was left out is included. For example a coloring of each vertex by relative position could result in a more complex sampling of the current prediction:

$$\phi : Q \times \Gamma \longrightarrow P \times \tau \times \gamma, \tag{18}$$

where $\gamma : N \longrightarrow \mathbb{N}^3, \gamma(x) = \langle i, j, k \rangle$ and $i, j, k \leq 255$.

D. Sequences of predictions

Recall the mapping ϕ (3.18) and let this generate the set $S = \{(\tau_v, \delta_v, l), l \in L\}$. Take a sequence of predictions from this set $s_1, s_2, ..., s_n, s \in S$. We have already defined an order structure of the orientations τ_v, for each individual stroke of this prediction, formally this means $(\tau_v, l) \subset s, s \in S$.

Further we can derive additional transformations through subsequent mappings. Let

$$\phi^1 = \{s_i \mid \overline{xy}_{j,s_{i-1}} = \overline{xy}_{i,s_i}, \forall i \in \mathbb{N}, x_{s_i} \in S \cap L\} \tag{19}$$

be the set resulting from mapping the set of all predictions by imposing the constraint that the endpoints of the lines be contiguous.

Theoretically there could be any number of such transformations. A single additional mapping that might be of interest is provided, ϕ^2, in which the longest common subsequence (LCS) algorithm is used to introduce two additional tunable parameters: *k* and *r*, which are the chosen subsequence length and the number of repetitions.

$$\begin{aligned} \phi^2(k, r) &= f(g(S^1)) \\ &= f(s_i \mid s_i \in \phi^1(S)) \\ &= \{s_i \mid m_{l_{s_i}} = m_{l_{s_{i+r}}}\} \end{aligned} \tag{20}$$

having $\| S^3 \| = m$, where m_i is the slope of the line.

Then, the problem has a set of mappings ϕ^i, that satisfy the constraints C:

$$C = \begin{cases} c_1, & \text{The endpoints are connected.} \\ c_2, & \text{They are formed by repetitions of the LCS.} \end{cases} \tag{21}$$

IV. Tilings

A. Motivation

The framework does not generate the tilings by itself, instead it uses a parser that scans images of tilings to detect the vertices of the polygons. For future work, more can be achieved by having the compositions be constructed because it would offer the system information about the evolution of the construction at each step. The mathematical context of scanned outcomes from the process of construction is still important to the study. The scene will be perceived differently depending on the observer's experience with recognizing the category to which the composition belongs to. The interaction with the screen and the predictions themselves are also constructions, or subsets of the complete construction.

The expert system that decides which is the next stroke to predict will base its decision on compositions of regular polygons which form tilings. The tilings are already categorized and have been an object of mathematical study for a long time. In order to elicit the subject's pattern recognition abilities, the game engine has a few advantages, despite having the same input modes as the user. First, it is able to flawlessly construct any tiling. Second, it is able to react much faster than the user. The game could be configured where the recommender constructs whole blocks of the tiling at once, which would in turn cover a large surface area of the screen which could be optimized to maximize learning and minimize session time for a user.

B. Tilings

The representation of tilings is a complex topic. Usually there is a diagrammatic language or a sequence of symbols to denote a tiling. We refer to the sequences as notations, which are further described.

The fundamental building block is a polygon, or sometimes the vertices and edges of these polygons, in Euclidean space. Regular tilings involve regular polygons as the building blocks. There are specific operations done iteratively to form a composite of the fundamental blocks. These can be rotation and reflection operations, for example. A construction where every polygon has a neighboring one at every edge is called a tesselation.

It is possible that some notations are too restrictive to produce every possible configuration. Cundy and Rollet's notation has some problems regarding the ability to generate unique tesselations because of ambiguity. [12]

There are many classifications of the possible ways of tiling the Euclidean plane, and this has been a subject of scientific debate for decades, before 1987 when Grünbaum and Shephard published their work. [13].

To every tiling corresponds its dual representation, a possible coloring, with other properties and particularities. Considering the dual graph where the centroid of each polygon is a vertex, and these nodes are connected. When the resulting tiling is identical to the original one, the object is called self-dual. The properties of tilings is a concern of representation theory, and it can highlight limitations of the expected interactions with a user, and explain which configurations can be the most promising for an interactive game.

Closely related figures are ones produced by the plane crystallographic groups, plane symmetry groups, or wallpaper groups.

C. Tiling grammar

Tiling grammars are proposed especially for a subset of tiling tasks where the regular polygons don't have to fill the Euclidean plane or $\mathbb{E}^3$ space. The only constraint is that the length of the connecting faces must be the same. A sequence of the grammar's symbol can be generated simply from cursor movements, and by operating on with predefined rules the grammar's symbols, the resulting composition might be rewritten or compressed.

It is possible to interpolate the drawn curve to a Bézier curve or another type of interpolated spline, which also normalizes it and removes noise, and use the curve to match the edges on the dual space of the history of tiled compositions. Graphically the size of a tiling composition can be adjusted to please the user and provide feedback at every step of the construction.

Suppose the left grammar,

$$\begin{gathered} G = \{N, \Sigma, P, S\}, \\ \Sigma = \{i \in \mathbb{N}\}, \end{gathered} \tag{22}$$

and an automaton that accepts pairs of two symbols X_k, X_{k+1} of input length $2 \cdot n$. An input sequence that looks like $0, 6, 1, 3, 2, 4$ will first produce a central hexagon, then attach a triangle on edge 1, and a square on edge 2 of the triangle. There are already formal definitions of such processes, but this example is shown because the programmer has the freedom to design any such equivalence in order to produce a set of tilings.

Following, each symbol is expanded with a set of symbols from another grammar,

$$G' = \{N', \Sigma', P', S'\}, \tag{23}$$

and the input $\forall X_i \in \Sigma$ is rewritten to :

$$\begin{gathered} X_j^{''} \in \{0X_1, 6X_2, 1X_3, \dots X_i X_j' \mid X_i \in G, X_j' \in G'\} \\ X_j^{''} \in G' \cup G. \end{gathered} \tag{24}$$

The symbols $X_j^{''}$ are equivalent to a decision support feedback process that is given to the user at the insertion of each symbol in the input tape. An insertion of the symbol corresponds to one of the dual transitions $\phi^{\dagger}$ defined in equation (3.16), which is paired to our now-defined grammar of walks on edges. The final point is coupling the decision support symbols $X_j^{''}$ to a data structure such as a graph. Any system that traverses the vertices of the figures generated can be viewed as a graph. The graph of transitions on the geometrical structure can transition between walks on edges, and walks on vertices of the figures, as long as the grammar allows it. Such a graph of vertices generated by centroids, edges or vertices, or any mapping of the components of the tilings to a set of points, can belong to a symmetry group like the Coxeter groups. Depending on the number of options available during a walk for the next transition, the system can decide the number of faces of the adjacent polygon as the tiling is generated. A transition can be viewed as adding a constraint to the space of possible polygons connected to the current one as the tiling is generated. Some of the faces can be constrained with the objective that the queries do not run into future self-collisions, or to compose polygons so that the construction will finish by running out of options in a number of steps. Ultimately, these compounds can be logged and rewritten or searched by pattern matching, or a feature search.

V. Annotating the EEG signals

The EEG signal is a time series signal of a given sampling frequency. For designing a biofeedback application, one must first clearly specify what are the relevant parameters of the signal that can influence the objective of the application. According to each individual subject, the application should detect what signals the brain responds to when faced with the stimuli, determine the target signal, and attempt to drive the signal into the desired parameters.

The EEG signal unlike the events happening on the whiteboard is continuous in the time axis. The first step in the pipeline is to reduce the quantity of the signals that we process. A time interval is chosen that is located in temporal proximity of each event rendered on the whiteboard. Such a time window can be seen in Fig. 2.

Recall equation (10) giving the strokes the property of time ordering. Taking the endpoints of the time interval we can gather as many time windows as there are channels.

One can consider a radius from the midpoint of an edge or centroid of a polygon and determine connected components of vertices on the rendered point cloud. According to the complexity of each connected component and the distance, angle, and regularity of each component, extra information becomes available to be used. An interesting measure of complexity was already defined using the LCS algorithm, in equations (20), (21). When considering strokes, the geometric information of the complexity of the drawing can be associated to the time windows that are extracted.

Another method of clustering EEG signals is by comparing their frequency bands. A signal can be understood through a

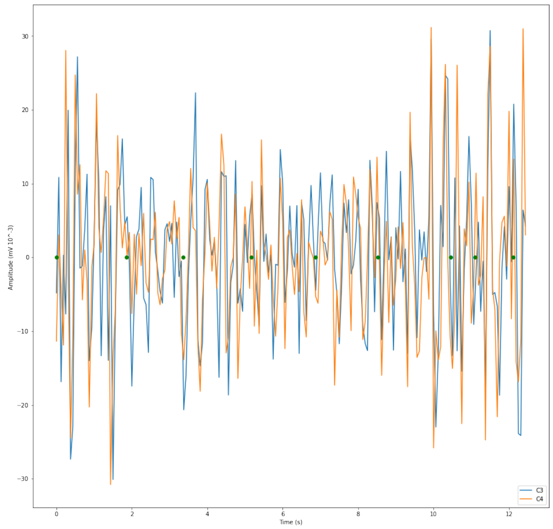

Fig. 2. Example of a time window from one of the experiments of the pilot study. Horizontal axis is time in seconds, vertical axis is amplitude in mV divided by 10^3. The green markers delimit the strokes that the user drew on the digital tablet. Electrodes chosen were C3 and C4.

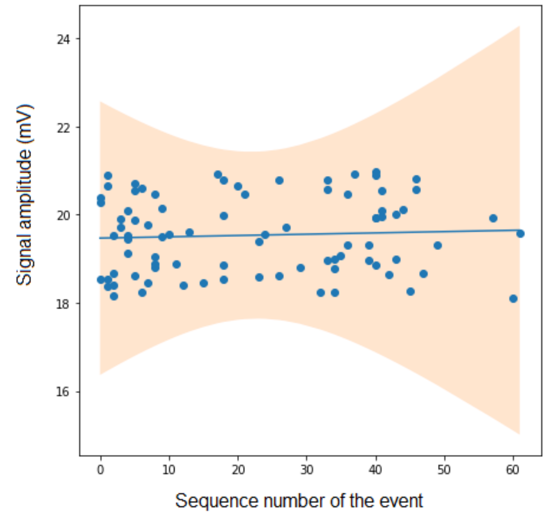

Fig. 3. Event related potentials. Horizontal axis describes the second in which it was detected. Vertical axis is the signal amplitude in mV describing the detected intensity of the spike. Data appears to follow a normal distribution with a few clusters. Electrodes chosen were C3 and C4.

spectral transformation as a set of values for the amplitudes of a particular frequency band. For example, the bands alpha, beta, gamma, delta, can have values associate with them in a specific time window which is shown to be similar in a time window across two channels that are symmetrically positioned on different hemispheres. The assumption can be that there is an inter-hemispheric synchronization, and can be placed in a data structure where the signals are ordered in terms of similarity. Then the context of the drawing board is compared across the most similar signals to understand which combination of connected components of tiles generate the most dominant responses in the user.

To conclude this section, the suggestions given provide the basis for investigating the geometric structure of the visual stimuli with the EEG time windows and enable the annotation of the time windows with contextual information about a structure generated by rotations and reflections.

VI. Results

A pilot study was conducted that consisted of 2 subjects who performed a total of 6 game sessions with randomized parameters. The randomized parameters include the choice of drawing to be constructed, which can be sampled from a pool of two drawings. Another parameter is given by the possibility of the participant choosing a single or batch of multiple random recommendations before continued interaction with the pen.

The combined data from all the sessions were visualized and a threshold for detecting event-related potentials (ERPs) was set at 1800mV positive or negative signal amplitude. By joining the data from all the sessions we were able to remark that there are naturally occurring clusters of ERPs towards the beginning and middle of the session, at times 5s and 38s, displayed in Fig.3 by a regression plot.

The most frequent spikes across all 6 sessions can also be visualized as a box plot like in Fig. 4 which evidences the most variable clusters of ERPs when projected on the X axis.

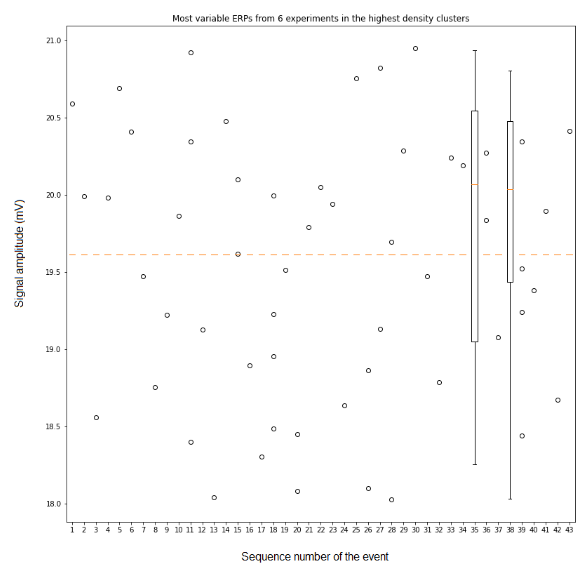

Fig. 4. Horizontal axis describes the order of the ERP triggered by an event that took place on the screen. On the Y axis is the value of the measurement, the minimum being the threshold for detection. The circles are the ERPs, and the boxes are the variability of 2 of the largest clusters of ERPs. The maximum value is 43 and was chosen to be the minimum across all sessions.

VII. Discussion

An explanation of why there are clusters at approximately the same stages across the sessions could be that the tilings naturally induce particular strategies that the subjects choose to follow. The natural processes ongoing in the brain of an individual could elicit responses by themselves, but by identifying the times most related to the visual feedback, the probability that a response is related to the ongoing activity is increased.

It has to be decided how the current stroke relates to the existing strokes on the drawing, because the previous ones will persist on the screen. The tilings naturally provide constraints on what are the trajectories that the subject can draw with the pen, because it is likely that a subject will choose from a set of similar game strategies, otherwise there is no progress in the task. For future work, it should be investigated how similar two visually distinct positions of a session actually are. A simple measure of relatedness is considering the slope and creating clusters by slope, to analyze how the signals change according to the slope of the prediction. Another question is what are the closest strokes to the last one that was rendered on the screen.

A remarkable feature of considering the discrete strokes by associations with EEG signals is that it is possible to perform a topological sort on the data. Because the strokes can be viewed as points on a planar graph, multiple ways of achieving a topological sort of this graph also induces a sorting of the associated EEG time windows. Formally this is seen as a bipartite graph coupling problem, where the connected components of the graph of edges comprising the tilings are matched to the graph of EEG time windows. By performing the topological sorting, it is possible to not only make cross-channel correlation analysis but also atemporal correlation analysis. Disregarding the order in which the strokes are drawn, it is possible to see if a particular signal does not depend on the previous strokes even if they are visible on the screen.

A concrete example of properties of the data that an expert system might employ is to use the concept of inter-hemispheric synchronization. The EEG electrodes are labelled by spatial position on the scalp. If two signals are considered similar on symmetric electrodes situated on different hemispheres, for example T5 and T6, by comparing the spectral power amplitude on a frequency band, the game might choose a future behavior that accounts for this. If otherwise the inter-hemispheric synchronization is the desired outcome, then the game engine could interact in such a way that could evoke a response in one of the hemispheres which could induce a synchronization. In Fig. 5 there are 5 STFT spectrograms, one for each of 3 randomly selected channels, where a data analyst might be able to visually compare the similarity of two channels.

VIII. Conclusions

The experiments of the pilot study have shown that the discrete nature of individual pen strokes provide clear temporal bounds for generating time windows for the EEG signals. By considering the context of an individual pen stroke within a geometric structure defined mathematically, we can investigate whether an EEG signal can reveal how the human brain responds to pattern recognition tasks. A user profile of the subjects is constructed when they attempt to perceive the abstract representations of apparently complex visual stimuli.

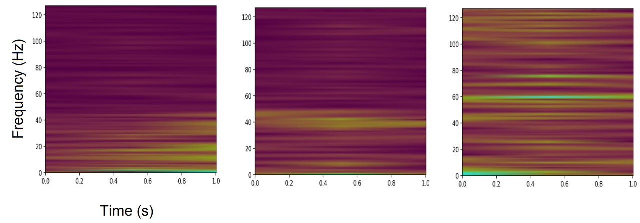

Fig. 5. A set of 3 spectrograms from different channels concatenated. Changes in amplitudes of particular frequency bands can also be used to correlate EEG data with events on the screen. Horizontal axis is time in seconds, vertical axis is frequency.

In the study we have given some examples of concrete hypotheses that can be tested using a hand-drawing biofeedback application. The actual space of hypotheses to test is very large and a large portion remains unexplored. The framework isolates particular configurations of objects rendered on the screen that are most likely to evoke responses.

References

[1] B. T. T. Yeo, F. M. Krienen, S. B. Eickhoff, S. N. Yaakub, P. T. Fox, R. L. Buckner, C. L. Asplund, and M. W. Chee, “Functional Specialization and Flexibility in Human Association Cortex,” *Cerebral Cortex*, vol. 25, no. 10, pp. 3654–3672, 09 2014.

[2] D. George and J. Hawkins, “Towards a mathematical theory of cortical micro-circuits,” *PLOS Computational Biology*, vol. 5, 10 2009.

[3] G. K. Anumanchipalli, J. Chartier, and E. F. Chang, “Speech synthesis from neural decoding of spoken sentences,” *Nature*, vol. 568, pp. 493–498, 2019.

[4] R. D. Pascual-Marqui, “Standardized low-resolution brain electromagnetic tomography (sloreta): technical details.” *Methods and findings in experimental and clinical pharmacology*, vol. 24 Suppl D, pp. 5–12, 2002.

[5] D. Yoshor, W. H. Bosking, G. M. Ghose, and J. H. R. Maunsell, “Receptive Fields in Human Visual Cortex Mapped with Surface Electrodes,” *Cerebral Cortex*, vol. 17, no. 10, pp. 2293–2302, 12 2006.

[6] B. M. J. Robinson Amanda K., Venkatesh Praveen, “Very high density eeg elucidates spatiotemporal aspects of early visual processing,” *Scientific Reports*, vol. 7, 2017.

[7] S. Mallat, “A theory for multiresolution signal decomposition: The wavelet representation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, pp. 674–693, 1989.

[8] V. Turri, “What is explainable ai?” Carnegie Mellon University’s Software Engineering Institute Blog, Jan. 17, 2022. [Online]. Available: http://insights.sei.cmu.edu/blog/what-is-explainable-ai/

[9] P. Xu, Y. Huang, T. Yuan, K. Pang, Y.-Z. Song, T. Xiang, T. M. Hospedales, Z. Ma, and J. Guo, “Sketchmate: Deep hashing for million-scale human sketch retrieval,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[10] H. Morita, D. Kawahara, and S. Kurohashi, “Morphological analysis for unsegmented languages using recurrent neural network language model,” in *EMNLP*, 2015.

[11] T. Shibata, D. Kawahara, and S. Kurohashi, “Neural network-based model for japanese predicate argument structure analysis,” in *ACL*, 2016.

[12] V. Gomez-Jauregui, H. Hogg, C. Manchado, and C. Otero, “Gomjau-hogg’s notation for automatic generation of k-uniform tessellations with antwerp v3.0,” *Symmetry*, vol. 13, no. 12, 2021.

[13] B. Grünbaum and G. C. Shephard, “Tilings by regular polygons,” *Mathematics Magazine*, vol. 50, no. 5, pp. 227–247, 1977.